FIRM AFS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: Daniel Blaukopf, et al.

Serial No. 09/963,435

Filed: September 27, 2001

For: Method and Protocol for Mediating Communication Between Software for Applications

§ Group Art Unit: 2152
§
§ Examiner: Chankong, Dohm
§
§ Atty. Dkt. No.: 5681-78901
§ P5399

CERTIFICATE OF MAILING
37 C.F.R. § 1.8

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below:

Robert C. Kowert
Name of Registered Representative

March 7, 2006
Date

Signature

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed January 3, 2006, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

03/14/2006 DEMMANU1 00000047 501505 09963435
01 FC:1402 500.00 DA

I. REAL PARTY IN INTEREST

As evidenced by the assignment recorded at Reel/Frame 013071/0748, the subject application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-20 stand finally rejected. The rejection of claims 1-20 is being appealed. A copy of claims 1-20 is included in the Claims Appendix herein below.

IV. STATUS OF AMENDMENTS

An amendment subsequent to the final rejection was filed on November 21, 2005. According to the Advisory Action of December 19, 2005, this amendment was entered by the Examiner. The copy of the claims included in the Claims Appendix herein below reflect entry of this amendment.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed toward a method of communicating functions or event notification between two applications. The additional requirement of converting platform independent code, such as Java, into native processor instructions may cause code written in platform independent languages to run less efficiently and require greater resources than code performing the same functionality in a processor's native code. Offloading some operations to concurrently running applications written in a processor's

native language may speed up the performance of applications written in platform independent languages. *See, e.g.*, page 1, line 20 – page 2, line 4.

The method recited in independent claim 1 includes a first application launching a second application. As described starting on page 4 and page 6 of Appellants' description, a first application may cause a second application to be launched. For example in one embodiment, the first application may be compiled or linked to a mediation module that may issue a command causing the operating system to produce an instance of the second application. The launching of the application includes the first application passing an event port number and a command port number to the second application. For example, a first application may launch a second and provide the second application, by means of command line arguments in one embodiment, with the numbers of two ports, such as TCP/IP ports to be used for communication between the two applications. In some embodiments, the port numbers are termed a command port and an event port. The ports numbers are stored in a memory accessible to the second application. For instance, the second application may store the command port number and the event port number for future reference in an accessible location. *See, e.g.*, FIGs. 1-3; page 2, lines 6-13; page 4, lines 12-21; page 4, line 25 – page 5, line 3; page 5, line 25 – page 6, line 14; page 7, line 25 – page 8, line 8; page 8, line 24 – page 9, line 11.

Independent claim 12 is directed to a computer accessible medium including instructions and connected to a processing unit. When the processing unit executes the instructions, a first application launches a second application. As with the method described above regarding claim 1, when launching the second application, the first application passes to the second application an event port number and a command port number that are stored in a memory accessible to the second application. Please refer to the discussion of claim 1 above for a more detailed description. *See, e.g.*, page 3, line 21 – page 4, line 3.

Independent claim 20 is directed to a device including a processor and a memory coupled to the processor. The memory includes program instructions configured to

implement a first application launching a second application. As with claims 1 and 12, described above, the launching of the second application includes the first application passing an event port number and a command port number to the second application and the port numbers are stored in a memory accessible to the second application. Please refer to the descriptions of claims 1 and 12 above for a more description. *See, e.g.*, page 3, line 13 – page 4, line 3.

The summary above describes various examples and embodiments of the claimed subject matter; however, the claims are not necessarily limited to any of these examples and embodiments. The claims should be interpreted based on the wording of the respective claims.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 12 and 20 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Aldred et al. (U.S. Patent 5,719,942) (hereinafter "Aldred") in view of Raynak et al. (U.S. Patent 5,680,549) (hereinafter "Raynak").

2. Claims 2-6, 8-11, 13-17 and 19 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Aldred and Raynak, and in further view of Simonoff et al. (U.S. Patent 6,005,568) (hereinafter "Simonoff").

3. Claims 7 and 18 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Aldred, Raynak and Simonoff, and in further view of Jalili et al. (U.S. Patent 5,423,042) (hereinafter "Jalili").

VII. ARGUMENT

First Ground of Rejection:

Claims 1, 12 and 20 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Aldred in view of Raynak. Appellants traverse this rejection for at

least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 12 and 20:

Regarding claim 1, contrary to the Examiner's assertion, Aldred in view of the Raynak does not teach or suggest a first application launching a second application, where the launching of the second application includes the first application passing an event port number and a command port number to the second application. Instead, Aldred specifically teaches the use of support system software together with call manager applications to establish, configure, and manage communication channels between applications, especially between applications executing on different hardware nodes (Abstract; column 1, lines 52-60). Aldred teaches that groups of applications communicate by participating in named sharing sets. Aldred's call managers coordinate, monitor and manage the various share sets of applications. Aldred also teaches a support system and a software API through which applications interact with the call managers. Aldred's API includes functions for initiating and configuring communication between shared applications via channels and signals. However, none of the teachings of Aldred describe a first application launching a second application, where the launching of the second application includes the first application passing an event port number and a command port number to the second application.

The Examiner cites various portions of Aldred (column 5, lines 51-63; column 6, lines 39-49; column 7, lines 33-62; column 12, lines 57-61; and column 36, lines 3-52) that describe Aldred's channels and share sets. However, none of these cited passages describes passing event port numbers and command port numbers to an application as part of launching that application. Instead, the manner and method of initializing and configuring communication channels between applications described in Aldred does not include passing event port numbers and command port numbers as part of launching an application. To the contrary, Aldred explains the benefits of using the support system software when establishing and configuring communications channels. For example,

relying upon call managers and the support system software allows applications to be "aware" of, and to use, Aldred's system while avoiding the need to be involved in "call set-up or tear-down." Aldred teaches the benefit of providing clear separation of call management and application programming (Aldred, column 26, lines 62-67). Thus, Aldred specifically does not describe a first application launching a second application, where the launching of the second application includes the first application passing an event port number and a command port number to the second application.

Aldred states, "in order for an application instance to be allowed to communicate with the system, it must identify itself by issuing a register_app call" (column 35, lines 48-67). Aldred also teaches, "it is up to the launched application to use [the] register_app [function] to fully identify itself to the system" (column 36, lines 21-55). Aldred describes that adding a port to a channel includes a request from one application, which is sent via the support system as an unsolicited event to a second application, and a confirmation (or error) response routed back to the first application as a confirm event (Aldred, column 24, lines 39-51). Additionally, one of the benefits of Aldred's share sets, call managers and support system software is that data may be communicated across heterogeneous networks using passive nodes to route data between an application on one node and another application on another node (Aldred, column 2, lines 19-50; column 5, lines 41-50; column 19, lines 24-48). Nowhere does Aldred describe passing event port numbers and command port numbers to an application ***as part of launching that application***.

The Examiner also cites column 11, lines 27-39 and column 29, lines 8-19, where Aldred describes launching applications and refers to Aldred's "launch_app" API command. However, Aldred's launch_app function is used by applications to interact with, and request support services from, Aldred's call managers (*see*, Aldred, column 4, lines 27-43). Thus, an application wishing to launch another application uses the launch_app function to communicate the request to a call manager. The call manager forwards the request to a call manager executing on the appropriate node of Aldred's system. The second call manager may then launch the requested application (Aldred,

column 5, lines 51-63). The fact that Aldred includes a mechanism to launch applications does not teach or suggest passing an event port number and a command port number as part of launching an application. **Nowhere does Aldred describe passing event port numbers and command port numbers as part of launching an application via the launch_app API function**. In contrast, Aldred teaches that an application may issue a launch_app command and may be returned a limited use handle to the launched application that is "only valid in very restricted circumstances *until the launched application has registered with the support system*" (emphasis added, column 11, lines 34-36). **Thus, as noted above, Aldred very clearly teaches that ports, and therefore event port numbers and command port numbers, are only configured after an application has registered with the support system.**

The Examiner, in the Response to Arguments section, "believes it is reasonable to suggest that the parameters passed in Aldred's launch_app function would be the channel characteristics needed for launching and launched applications to communicate". The Examiner repeats this assertion in the Advisory Action. However, the Examiner's belief is completely unsupported by the actual teachings of the cited art, and can thus only be based on hindsight knowledge of Appellants' disclosure. In fact, **Aldred teaches away** from one application passing event port numbers and command port numbers as part of launching another application. As described above, Aldred's system already includes a very specific mechanism to initiate and configure ports and channels between applications that specifically does not include passing event port numbers and command port numbers as part of launching applications. Aldred clearly teaches the benefits of an application first registering with a call manager and joining a share set before initiating or configuring channels and ports. Rather than providing any motivation to modify Aldred's system to pass event port numbers and command port numbers as part of one application launching another application, **Aldred teaches away** from one application launching another application *and passing event port numbers and command port numbers as part of launching the other application*. Furthermore, it would not make sense to modify Aldred to bypass the share sets that are central to Aldred's system by passing event port numbers and command port numbers as part of launching applications.

Such a modification would not only be contrary to Aldred's specific teachings, it would remove the specific benefits of Aldred's sharing sets, call managers, and support system software.

The Examiner, in the Advisory Action, responds to this argument by repeating the assertion that Aldred's launch_app function might include command and event port numbers and again citing column 36, lines 24-53 of Aldred. However, as noted above, this passage of Aldred fails to mention sending command and/or event port numbers as part of one application launching another application. Moreover, the cited passage describes the "parameters" referred to by the Examiner as "a user specified string that is given to the launched application". The Examiner appears to be interpreting the "user specified string" as including command and event port numbers. However, nowhere does Aldred mention that a user of his system specifies command and event port numbers. Instead, as noted above, Aldred system includes a specific set of APIs allowing applications to setup command and event ports, including specifying port numbers.

The Examiner also asserts, both in the Final Office Action and the Advisory Action, that "the sending application is responsible for defining the channel characteristics", and that "modification of Aldred to include passing an channel and port information between applications does not change the principle of his invention". The Examiner cites column 1, lines 61-65. However, the Examiner has clearly mischaracterized Aldred's meaning regarding "the sending application is responsible for defining the channel characteristics" by stating instead "the sending application is responsible for establishing the channel between applications". *Defining the channel characteristics* and *establishing the channel* are two different actions. In fact, Aldred describes these characteristics in column 12, lines 57-64 as: "target application handle, channel set type and identifier, data class, maximum buffer size, user exit, node handle, quality of service, connect type, port event handler, user information". In Aldred's system, the sending application must provide the support system with this information in channel creation; but *it does not establish the channel itself.* Appellants also note that in creating the channel between two programs, i.e., the launched and launching programs, a

target application handle and a node handle are required by the support system. The Examiner speculates (erroneously) that "parameters passed in Aldred's launch_app function would be the channel characteristics needed for the launching and launched applications to communicate". However, the Examiner has not cited any portion of the art to support such a suggestion. As noted above, a target application handle and a node handle are required for the channel creation, and as stated at column 11, lines 27-39, are not available until *after* the application has been launched. Furthermore, the node handle is specified by the return data, which is returned *after the application has registered* with the support system (column 11, lines 29-31 and 36-39).

Furthermore, contrary to the Examiner's assertion in his Response to Arguments, the limited use handle does not disclose "that a channel has been already established between the applications". In fact, Aldred defines how the handle is implemented in column 36, lines 45-48: "This function [launch_app] is used to ask the system to start another program instance. IF the new application is started successfully then its instance handle is inserted in the target_application and returned to the calling application". Aldred has already defined the target_application as a pointer used by the system. Therefore, Aldred's system changing the value of a particular pointer, where that changing of value is communicated within the support system and not directly between the two programs, does not teach that a channel has been already established between the applications. And, as noted above, the newly launched application is not "allowed to communicate with the system" without registration. Furthermore, this communication with the system is required for the system to create a channel as in the API call function add_channel or create_channel (column 29).

Moreover, modifying Aldred's system to include passing an event port number and a command port number to an application as part of launching that application would change the principle of operation of Aldred's system. Aldred's system relies upon applications registering and utilizing both the call managers and the support system software via Aldred's API to properly initiate and configure communications between applications. Bypassing this system to send event port numbers

and command port numbers to applications, as part of launching those applications, would bypass Aldred's sharing set concept, which is essential to the operation of his system, and thus change Aldred's principle of operation. As discussed in M.P.E.P. § 2143.01, a rejection based on a modification that changes the principle of operation of a reference is improper.

Additionally, the Examiner asserts, in the Advisory Action, "Aldred discloses that the register function merely allows the support system to be aware of the application", citing column 36, lines 9-16. **However, the cited passage makes no such statement.** Instead, the cited passage describes how if no call manager currently exists the issuer of the register_app call either becomes the call manager or terminates. The Examiner has misrepresented the true teachings of Aldred.

Furthermore, the Examiner has stated that nowhere does Aldred declare that ports are only configured after an application has registered with the support system. Apparently the Examiner has overlooked the fact that Aldred specifically discloses that: "In order for an application instance to be allowed to communicate with the system, it must identify itself by issuing an register_app call. This call must be issued prior to any other calls from this instance, otherwise the calls will fail." Additionally, Aldred teaches, "it is up to the launched application to use an register_app to fully identify itself to the system" (column 35, lines 47-67). Clearly, Aldred's registration allows the newly launched application to interact with the system and is <u>not</u> simply limited to "allow other applications to know that it has been launched", as the Examiner contends.

The Examiner also relies on Raynak to teach one application launching another application and passing communication parameters as part of launching the launched application. Raynak a teaches method for transferring a network connection to an external host to a second application. Raynak teaches that an Information Manager (IM) application may launch a secondary application and pass a communication port number (as well as a baud rate) as command line parameters when executing the secondary application. The Examiner cites FIG. 4, column 1, lines 11-24 and column 6, lines 32-57

of Raynak. However, Raynak does not teach or suggest passing a command port number and an event port number. Instead, Raynak only teaches passing a communication port ID (e.g. COM1, COM2, etc) and a baud rate. Thus, the Examiner's combination of Aldred and Raynak does not teach or suggest that the launching of the second application includes the first application passing an event port number and a command port number to the second application.

Additionally, the Examiner's combination of Aldred and Raynak is improper because, as noted above, Aldred teaches away from passing a command port number and an event port number when launching a second application. M.P.E.P. 2144.05.III states that a "prima facie case of obviousness may also be rebutted by showing that the art, in any material respect, teaches away from the claimed invention". See also, *In re Geisler*, 116 F.3d 1465, 1471, 43 USPQ2d 1362, 1366 (Fed. Cir. 1997). Additionally, "[a] reference may be said to teach away when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the applicant." *In re Gurley*, 27 F.3d 551, 553 (Fed. Cir.1994) (*emphasis added*). As noted at M.P.E.P. § 2141.02, paragraph 12, a "prior art reference *must* be considered in its entirety, i.e. as a whole, including portions that would lead away from the claimed invention" (italics added, underlining in original). One of ordinary skill in the art, upon reading Aldred's teachings regarding the use (and benefits) of the call manager and joining a share set before initiating or configuring communications channels and ports, as described above, would clearly be discouraged from following Raynak's teachings regarding modifying an initialization file to include command line parameters specifying a communications port and baud rate. Further, one reading the teachings of Raynak regarding the use of command line parameters to specify a communication port and a baud rate would clearly be led in a direction away from the teaching of Aldred regarding the use of a call manager and a joining a share set. The respective teachings of Aldred and Raynak pertain to very different methodologies that clearly teach away from one anther. As such, the Examiner's proposed combination of Aldred and Raynak is improper.

Furthermore, the Examiner has failed to provide a proper motivation for combining the teachings of Aldred and Raynak. The Examiner simply states that it would have been obvious without giving any reasons why one would be motivated to do so (Office Action dated October 3, 2005, page 9, lines 10-12). An obviousness rejection that lacks evidence of a suggestion or motivation for one of skill in the art to combine prior art references to produce the claimed invention is defective as hindsight analysis. In addition, the showing of a suggestion, teaching, or motivation to combine prior teachings "must be clear and particular . . .. Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence'." *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination.

As argued above, the rejection of claim 1 is not supported by Aldred and Raynak, whether considered singly or in combination. Thus, for at least the reasons provided above, Appellants submit that the rejection of claim 1 is unsupported by the cited art and the Examiner has failed to establish a *prima facie* rejection. Similar remarks also apply to claims 12 and 20.

Second Ground of Rejection:

Claims 2-6, 8-11, 13-17 and 19 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Aldred and Raynak, and in further view of Simonoff. Appellants respectfully traverse this rejection for at least the reasons presented above regarding their respective independent claims. Different groups of claims are addressed under their respective subheadings.

Claims 2-4, 8, 9-11, 13-15, and 19:

The rejection of claims 2-4, 8, 9-11, 13-15 and 19 are allowable for at least the reasons presented above regarding their respective independent claims.

Claims 5 and 16:

Regarding claim 5, Aldred in view of Simonoff does not teach or suggest passing a function reference value through the command port connection. The Examiner cites column 24, lines 52-61 of Aldred. However, this portion of Aldred is referring to how applications can make asynchronous calls to Aldred's support system software API by including a reference identifier allowing the application to issue command to obtain the status of, or to cancel, an asynchronous API call. The cited passage does not describe passing a function reference value through a command port connection, as suggested by the Examiner. The reference identifier mentioned in the cited passage is not a function reference value that is sent through a command port connection. Instead, it is merely an identifier to allow a calling application to check on the status of an asynchronous API call.

In the Response to Arguments, the Examiner asserts that Aldred's reference identifier passes through the command port. However, in Aldred, the status of an API call is handled between the application issuing the call and the support system, and not between the launching and launched applications. The support system is only inherent in Aldred's system, and therefore, when communication occurs between only an application and the support system, Aldred's command port does not equate to that of the instant application. The Examiner does not refer to Simonoff in the rejection of claim 5. Aldred does not disclose passing a function reference value *through a command port*. Nor does Simonoff overcome Aldred's failure to teach or suggest passing a function reference value through a command port connection. Thus, the rejection of claim 5 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks apply to claim 16.

Claim 6:

Regarding claim 6, Aldred and Raynak, in further view of Simonoff, does not teach or suggest passing a function parameter through the command port connection.

The Examiner cites column 24, lines 39-42 of Aldred. However, this passage of Aldred describes how Aldred's system handles an application's request for a service. Specifically, Aldred teaches that an application requests a service and supplies the appropriate parameters. Another application, supplying the service, is made aware of the request through an "unsolicited event which appears as an indication" to the second application. The response from the second application is routing back to the requesting application as a "confirm event." The cited passage does not mention passing a function parameter through a command port connection, but instead teaches the use of unsolicited events to request a service and to deliver responses. Elsewhere (column 7, lines 44-62) Aldred teaches three different types of port connections: event, command and null ports. The passage cited by the Examiner in the rejection of claim 6 clearly refers to issuing events via event ports. The cited passages do not mention any command port connection.

In the Response to Arguments, the Examiner reasserts that the parameters are submitted through the command port because "command ports allow the application to drive the receipt or supply of *data* to the port". The Examiner is clearly redefining parameters of data using hindsight speculation, which is improper. Furthermore, the Examiner cites columns 35 and 36 and asserts, "functions are submitted through the command port". The cited text does not disclose command ports at all; in fact, columns 35 and 36 repeatedly disclose events and event handlers associated with the functions that would be communicated through the event port. The cited section in column 24 clearly defines the process associated with the "appropriate parameters" through events. Events are passed along the event port. The Examiner speculates that parameters are passed along the command port, but nowhere provides evidence or an example to support this. Thus, the rejection of claim 6 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claim 17.

Third Ground of Rejection:

Claims 7 and 18 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Aldred, Raynak and Simonoff, and in further view of Jalili et al. (U.S.

Patent 5,423,042) (hereinafter "Jalili"). Appellants respectfully traverse the rejection of claims 7 and 18 for at least the reasons presented above regarding their respective independent claims.

Further regarding claim 7, the Examiner's combination of Aldred, Raynak, Simonoff and Jalili fails to teach or suggest passing a value of a memory location for storing results of a function triggered by the passing of the function value. The Examiner relies on Jilili, citing the Abstract and column 10, lines 33-48 of Jalili. However, Jalili does not teach passing a value of a memory location for storing results. Instead, Jalili teaches that a client uses a SET FUNCTION DATA request to supply the arguments for a requested function and a GET FUNCTION DATA to obtain the results of the function. When using both the SET FUNCTION DATA and the GET FUNCTION DATA, a client in Jalili's system sends the server the function name, type and instance to identify the requested function and results (Jalili, column 9, lines 41-64). Jalili also teaches that the server, when executing a requested function, uses the values of the state 288 field of the exec_table. Jalili states, "[t]he results 289 value is a pointer to a server memory space where the results of running the function will be stored" (Jalili, column 10, lines 40-43). However, Jalili does not teach passing a value of a memory location for storing results. Instead, Jalili teaches that after performing the requested function, the results, "which are stored in the memory space pointed to by the results 289 value, are send back to the client" (Jalili, column 10, lines 43-48). Nowhere does Jalili mention passing a value of a memory location for storing results of a function triggered by the passing of the function value. Instead, as noted above, Jalili teaches passing the results of the executing a function.

Aldred, Raynak and Simonoff fail also fail to teach or suggest passing a value of a memory location for storing results of a function triggered by the passing of the function value and thus fail to overcome the above-noted deficiencies of Jalili. The Examiner combination of Aldred, Raynak, Simonoff and Jalili clearly fails to teach or suggest passing a value of a memory location for storing results of a function triggered by the passing of the function value. Therefore, for at least the reasons presented above, the

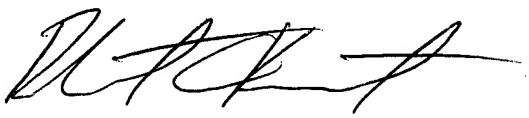
rejection of claim 7 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claim 18.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-20 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-78901/RCK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: March 7, 2006

IX. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A method of communicating function calls or event notification between two applications, said method comprising:

a first application launching a second application, wherein the launching of the second application includes the first application passing an event port number and a command port number to the second application, wherein the port numbers are stored in a memory accessible to the second application.

2. The method according to claim 1, further comprising the second application connecting a TCP/IP client socket to the event port.

3. The method according to claim 2, further comprising connecting a TCP/IP client socket to the command port.

4. The method according to claim 3, storing the connection parameters of either client socket.

5. The method according to claim 3, further comprising passing a function reference value through the command port connection.

6. The method according to claim 5, further comprising passing a function parameter through the command port connection.

7. The method according to claim 5, further comprising passing a value of memory location for storing result of a function trigger by the passing of the function value.

8. The method according to claim 2, further comprising passing an event notification tag through event port connection.

9. The method according to claim 8, further comprising checking the event port for an event notification tag.

10. The method according to claim 9, further comprising checking the command port in response to receiving an event notification tag.

11. The method according to claim 9, passing through the event port connection an event port notification tag relating to the completion of a function.

12. A computer accessible medium containing instructions and operatively connected to a processing unit, such that when said processing unit executes the instructions a first application launches a second application, wherein in launching the second application, the first application passes to the second application an event port number and a command port and the port numbers are stored in a memory accessible to the second application.

13. The computer accessible medium according to claim 12, further containing instructions that when executed by said processing unit cause the second application to connect a TCP/IP client socket to the event port.

14. The computer accessible medium according to claim 13, further containing instructions that when executed by said processing unit causes the second application to connect a TCP/IP client socket to the command port.

15. The computer accessible medium according to claim 14, further containing instructions that when executed by said processing unit cause the connection

parameters of either client socket to be stored in memory.

16. The computer accessible medium according to claim 13, further containing instructions that when executed by said processing unit causes the first application to pass a function reference value through the command port connection.

17. The computer accessible medium according to claim 16, further containing instructions that when executed by said processing unit cause the first application to pass a function parameter through the command port connection.

18. The computer accessible medium according to claim 16, further containing instructions that when executed by said processing unit cause the first application to pass a value of a memory location for storing result of a function trigger by the passing of the function reference value.

19. The computer accessible medium according to claim 13, further containing instructions that when executed by said processing unit cause the first application to pass an event notification tag through event port connection.

20. A device, comprising:

a processor; and

a memory coupled to the processor, wherein the memory comprises program instructions configured to implement:

a first application launching a second application, wherein the launching of the second application includes the first application passing an event port number and a command port number to the second

application, wherein the port numbers are stored in a memory accessible to the second application.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.